Whitepaper

White Paper Automation AIS-Create Document Service v2.1

EU GDPR COMPLIANT

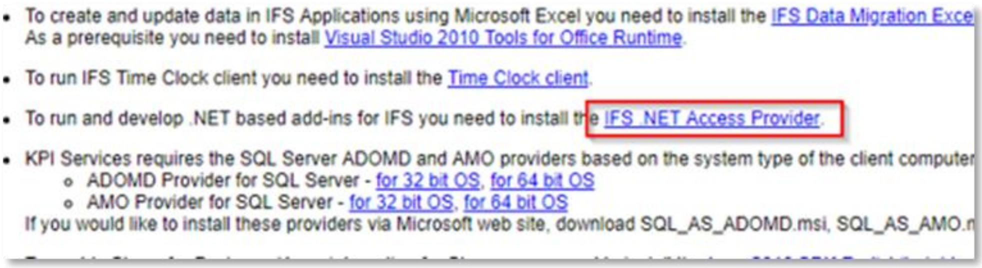
Updated: Date 2022-06-30

Contents

1 Infrastructure & technology

1.1 IFS Applications

Addovation Create Document Service is built on the recommended integration component delivered by IFS World called IFS .NET Access Provider normally found within the add-on section for a typical IFS installation:



- To create and update data in IFS Applications using Microsoft Excel you need to install the IFS Data Migration Exce
  As a prerequisite you need to install Visual Studio 2010 Tools for Office Runtime.
- To run IFS Time Clock client you need to install the Time Clock client.
- To run and develop .NET based add-ins for IFS you need to install the IFS .NET Access Provider.
- KPI Services requires the SQL Server ADOMD and AMO providers based on the system type of the client computer
  - ADOMD Provider for SQL Server - for 32 bit OS, for 64 bit OS
  - AMO Provider for SQL Server - for 32 bit OS, for 64 bit OS

  If you would like to install these providers via Microsoft web site, download SQL_AS_ADOMD.msi, SQL_AS_AMO.n

Figure 1 - IFS .NET Access Provider

The IFS Access Provider allows clients to communicate in the same way as IFS Enterprise Explorer in terms of:

- Authentication, users are managed by and in IFS Applications
- Security, by using IFS Applications to determine access to data and database objects such as API's or so-called business logic

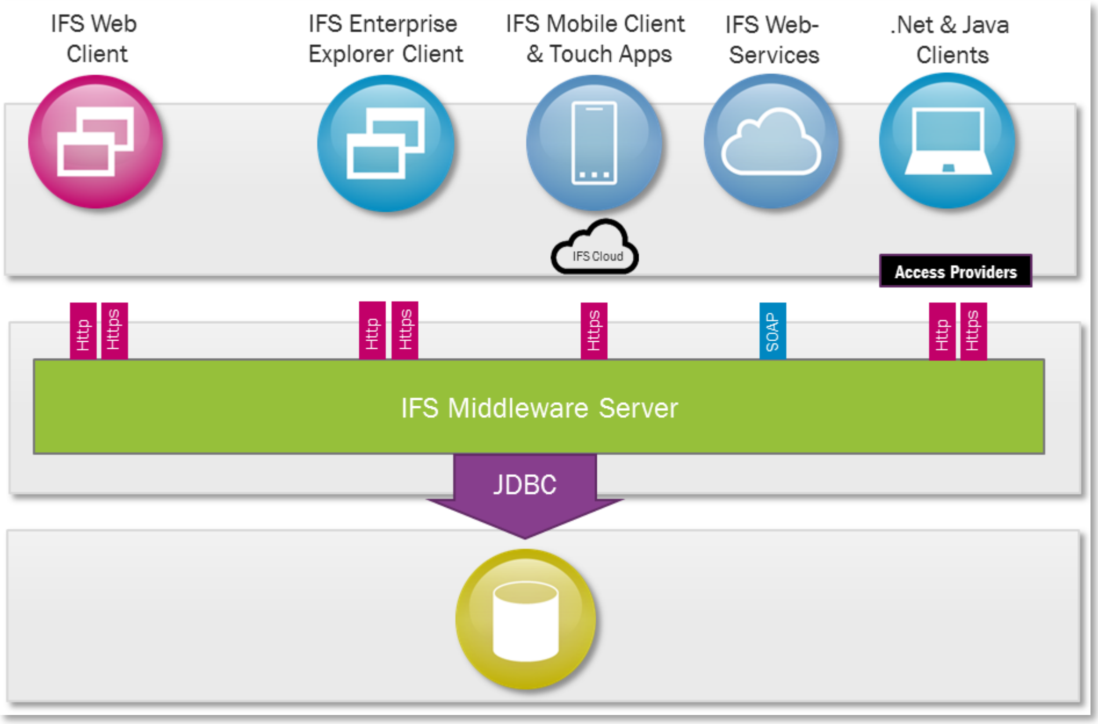


Figure 2 - IFS Application Architecture

More information about this can be found in documentation included with your local IFS Applications installation. Replace the server name (<yourIFSinstallation>) with your installation's path to read the following file.

Addovation Create Document Service is a service that creates documents (Word and PDF), based on a Word EDM template in IFS Applications. It can create a view copy of an existing Word document, but also merge information using a template prepared by Office Automation Suite server side.

1.2 IFS Cloud

IFS Cloud is very different from IFS Applications in terms of integration. Addovation Solutions for IFS Cloud communicates with IFS RESTful APIs called projections and authenticates using IAM.

One can read more about the concept here and full documentation for IFS Cloud is available at https://docs.ifs.com/techdocs/22r1/.

2 System requirements

The table below describes the system requirements for machines that is hosting the service on-premise:

System Requirements for Addovation Create Document Service	
Operating System	Windows 10 (32-bit or 64-bit) and above Windows Server 2019 (32-bit or 64-bit) and above
Memory	4 GB RAM or more
CPU	Intel/AMD (Dual core or more recommended)
.NET Framework	Full 4.6.2 or newer
IIS	10.0 or newer

*) Other Microsoft Windows versions might work but is unsupported.

**) Depending on load and volume it might be required to scale the machine differently.

3 Compatibility matrix

Addovation Create Document Service module is compatible with/requires the following operating systems, frameworks and applications:

SUPPORTS / REQUIRES	VERSION / BITNESS	CREATE DOCUMENT
WINDOWS SERVER 2012	32 and 64	✗
WINDOWS SERVER 2016	32 and 64	✓
WINDOWS SERVER 2019	32 and 64	✓
WINDOWS 7	32 and 64	✗
WINDOWS 8	32 and 64	✗
WINDOWS 10	32 and 64	✓
WINDOWS 11	32 and 64	✗
IFS APPLICATIONS	7	✗
IFS APPLICATIONS	8	✗
IFS APPLICATIONS	9 Upd 15	✓
IFS APPLICATIONS	10 Upd 9	✓
IFS CLOUD	21R2	✓
IFS CLOUD	22R1	✓

4 Limitations

Non-standard configurations in terms of authentication towards IFS Applications is not supported. Also, customizations that might affect standard API:s being used is not supported, for full list of used queries and API:s please contact Addovation if in doubt.

Installation of software requires writing to registry in local machine hive.

Addovation does not install any special views or packages in the database.
Only IFS storage using Oracle storage has been tested and verified.

For IFS Cloud, the service relies on IFS Cloud built in projections (non-integration) for creating new revisions, handling of statuses etc. Upon patching or updating it is required to re-test the solution and templates.

Customer is responsible for setting up and configure the following:

- IFS Connect
  - Specific queue.
  - Routing rules.
  - Routing address.
  - SOAP envelope.
  - Possible transformer.
- Event and custom action
  - PLSQL that holds logic to trigger application message.
- Possible custom button in Aurena to trigger event and action.
- Projection configurations to expose event execution.
- Integration / IAM
  - Integration user that has access to relevant information and objects.
  - Appropriate rights and permission sets to projections.
  - Basic data such as access to sites and companies.
- Document management
  - All basic data.
  - Classes and formats.
  - Access templates and rights.
- Object connections.
- IIS and support for WCF services.

4.1 IFS Cloud 21R2 / 22R1 specific limitations

In IFS document management, it is not possible to create both ORIGINAL and VIEW copy on same document revision. Either ORIGINAL or VIEW copy can be checked in, this is due to limitation in IFS Cloud.

In regards of datatypes, Addovation is limited to what datatypes IFS delivers via quick reports. The product only supports what IFS supports.

The following is not supported per se:

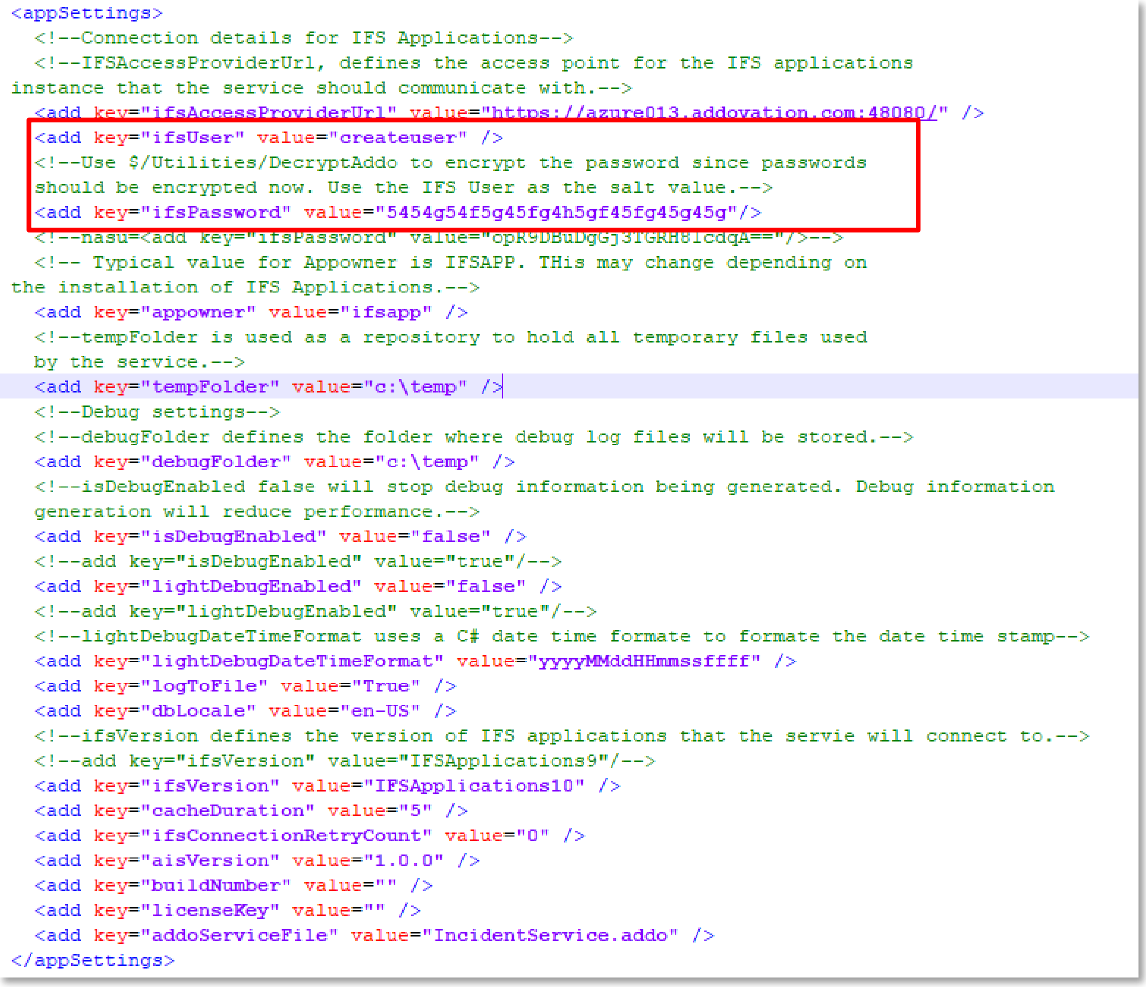
- AddoContentBlock.
- Merging documents into other documents.
- QR-/bar codes and fetching media library items.
- Generating hyperlinks.

5 GDPR compliance

Automation Create Document Service from Addovation obeys regulations controlled by GDPR and communicates in the same way as:

- For IFS Applications it communicates in same way as IFS Enterprise Explorer 9 and 10.
- For IFS Cloud it communicates against IFS RESTful services and IAM.

The information that is being stored is locally on the machine that are settings in the configuration file:

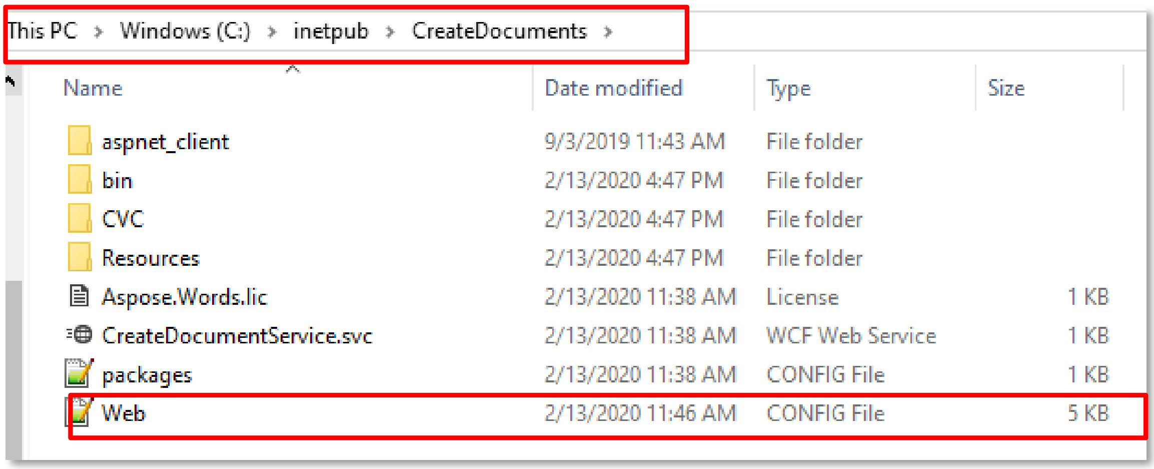


```
<appSettings>
  <!--Connection details for IFS Applications-->
  <!--IFSAccessProviderUrl, defines the access point for the IFS applications
instance that the service should communicate with.-->
  <add key="ifsAccessProviderUrl" value="https://azure013.addovation.com:48080/" />
  <add key="ifsUser" value="createuser" />
  <!--Use $/Utilities/DecryptAddo to encrypt the password since passwords
  should be encrypted now. Use the IFS User as the salt value.-->
  <add key="ifsPassword" value="5454g54f5g45fg4h5gf45fg45g45g"/>
  <!--nasu=<add key="ifsPassword" value="opR9DBuDgGj3TGRH81cdqA=="/>-->
  <!-- Typical value for Appowner is IFSAPP. THis may change depending on
the installation of IFS Applications.-->
  <add key="appowner" value="ifsapp" />
  <!--tempFolder is used as a repository to hold all temporary files used
  by the service.-->
  <add key="tempFolder" value="c:\temp" />
  <!--Debug settings-->
  <!--debugFolder defines the folder where debug log files will be stored.-->
  <add key="debugFolder" value="c:\temp" />
  <!--isDebugEnabled false will stop debug information being generated. Debug information
  generation will reduce performance.-->
  <add key="isDebugEnabled" value="false" />
  <!--add key="isDebugEnabled" value="true"/-->
  <add key="lightDebugEnabled" value="false" />
  <!--add key="lightDebugEnabled" value="true"/-->
  <!--lightDebugDateTimeFormat uses a C# date time formate to formate the date time stamp-->
  <add key="lightDebugDateTimeFormat" value="yyyyMMddHHmmssffff" />
  <add key="logToFile" value="True" />
  <add key="dbLocale" value="en-US" />
  <!--ifsVersion defines the version of IFS applications that the servie will connect to.-->
  <!--add key="ifsVersion" value="IFSApplications9"/-->
  <add key="ifsVersion" value="IFSApplications10" />
  <add key="cacheDuration" value="5" />
  <add key="ifsConnectionRetryCount" value="0" />
  <add key="aisVersion" value="1.0.0" />
  <add key="buildNumber" value="" />
  <add key="licenseKey" value="" />
  <add key="addoServiceFile" value="IncidentService.addo" />
</appSettings>
```

Figure 3 Configuration File

User credentials are encrypted and stored in the config file to communicate with IFS Applications and IFS Cloud.

Example in defined folder:



This PC > Windows (C:) > inetpub > CreateDocuments >

Name	Date modified	Type	Size
aspnet_client	9/3/2019 11:43 AM	File folder	
bin	2/13/2020 4:47 PM	File folder	
CVC	2/13/2020 4:47 PM	File folder	
Resources	2/13/2020 4:47 PM	File folder	
Aspose.Words.lic	2/13/2020 11:38 AM	License	1 KB
CreateDocumentService.svc	2/13/2020 11:38 AM	WCF Web Service	1 KB
packages	2/13/2020 11:38 AM	CONFIG File	1 KB
Web	2/13/2020 11:46 AM	CONFIG File	5 KB

Figure 4 Located Folder

Log file
If the end-user has enabled log on information, all the commands and PLSQL blocks will be stored in the file.

Purpose is to ease debugging for support personnel. Log file can easily be deleted. No passwords are being stored in log file.

6 Contact

For further inquiries, please contact your nearest Addovation office available at our web site: https://www.addovation.com/about-addovation/.

For existing customers, please use http://support.addovation.com.